**John T. Anderson**

# Engineering Note

**Date:** January 19,2000
**Rev Date:** January 19,2000

**Project:** D0 Central Fiber Tracker
**Doc. No:** A1000119

**Subject:** Test data generation in Analog Front End Board

## *Introduction*

The Central Fiber Tracker system consists of approximately 100,000 channels of fiber optic signals which are received and processed by Analog Front End (AFE) boards. Although the end intent is to provide a system which just takes the data without adjustment, in the process of getting to that point known test data will have to be generated at multiple points within the system. No method exists to drive known patterns of charge through the detector upon demand; although use of cosmic rays and low-energy beam will provide information to check the input cable plant, the Analog Front End must also be capable of generating test data upon demand.

## *System and Board Architecture*

198 AFE boards send data to some 64 Digital Front End boards over approximately 550 high-speed serial links. 300 of the links send 21 bits every 18.9 nsec, whereas the other 250 send 28 bits every 18.9 nsec, for an aggregate bandwidth of about 703 Gbits/sec, carried over 550 cables, each of which has 6 twisted pairs. Obviously some method of providing known data patterrns on demand is required to map out this system. All AFE boards are also connected to SVX Sequencer modules via a parallel cable interface, and all AFE boards are connected to MIL-STD 1553 for control and commands. A sketch of the system is shown in Figure 1.

## *Data Path Testing*

All data sent on the high-speed serial links is sourced by programmable logic devices (CPLDs) which, in normal operation, latch the data from the analog measurement circuits and send out all the bits over a set of seven ticks of the 53.104 MHz system clock. Thus, all boards are read out over a period of 132 nsec (one beam crossing). The use of CPLDs allows for reconfiguration of the system when required. In the most basic setups alternate programs may be loaded into the CPLDs which, in response to the clock, generate simple counts or shift patterns onto the links as opposed to real data. This is sufficient for the initial installation of the sytem when physical access to the boards is relatively easy. All CPLDs are the 'in-system programmable' variety which allows use of a personal computer or laptop and a simple cable connection to download new programs into the CPLD as desired. However, once the system is installed, access to the boards is not available and an alternate method of generating test data upon demand is required.

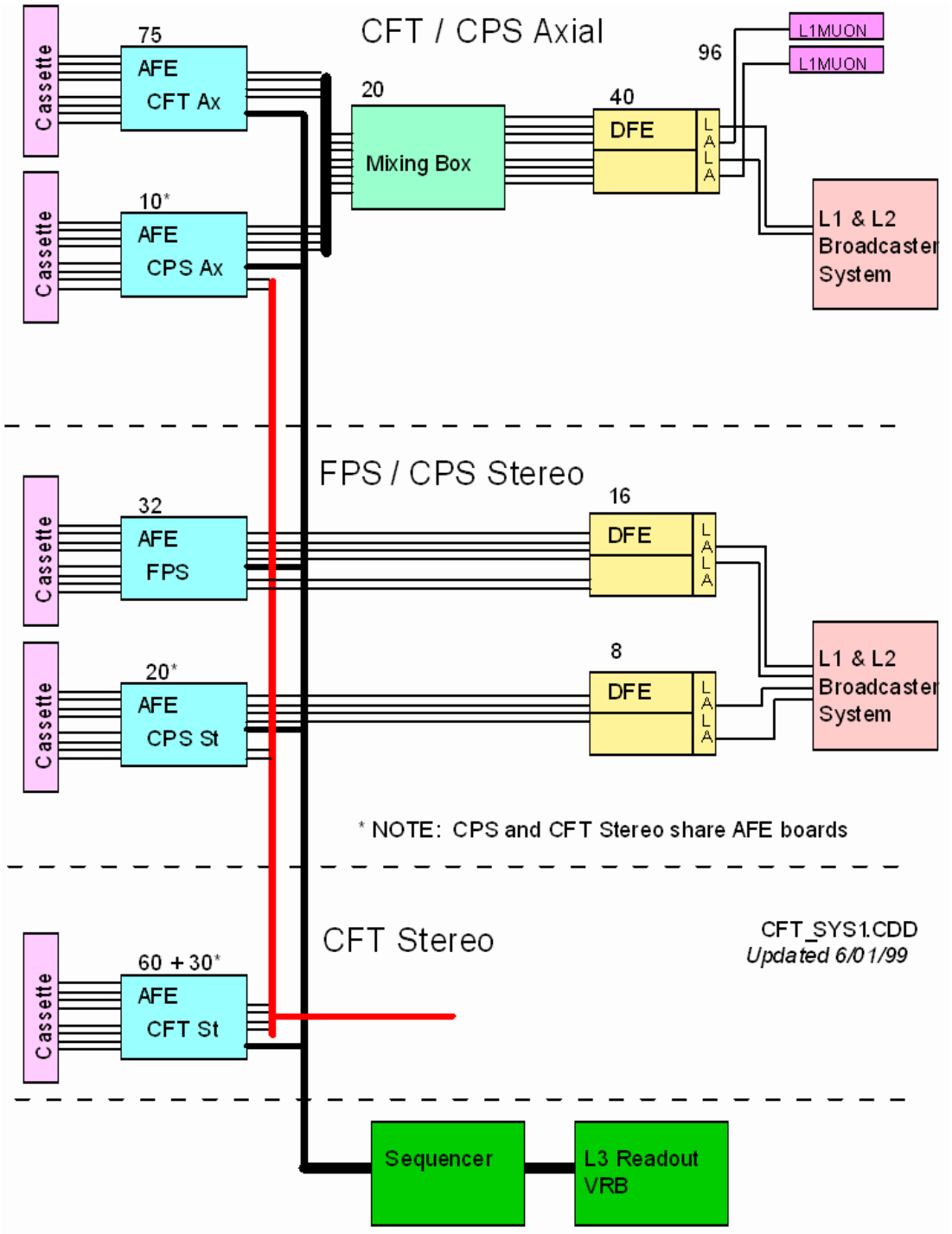
CFT / CPS Axial
Cassette
75
AFE
CFT Ax
Cassette
10*
AFE
CPS Ax
20
Mixing Box
40
DFE
LALA
96
L1MUON
L1MUON
L1 & L2
Broadcaster
System
FPS / CPS Stereo
Cassette
32
AFE
FPS
16
DFE
LALA
Cassette
20*
AFE
CPS St
8
DFE
LALA
L1 & L2
Broadcaster
System
* NOTE: CPS and CFT Stereo share AFE boards
CFT Stereo
CFT_SYS1.CDD
Updated 6/01/99
Cassette
60 + 30*
AFE
CFT St
Sequencer
L3 Readout
VRB

**Figure 1**

## *In-Place Test Pattern Generation*

Analysis of the clocking scheme used in the CFT system shows that beam is not delivered continuously. Every few microseconds a gap in the beam provides a time window during which no real data is available. This time interval is used by the analog conversion circuitry for resets. A signal named SYNC_GAP is provided to each AFE board by the SVX Sequencers which is asserted during these 'no-beam' times. This signal is provided to the Clock Generator circuit of the AFE, which creates two signals LD_SHFT and LD_TEST that control the data CPLDs. When the SYNC_GAP is not asserted, LD_SHFT goes low for one clock cycle out of every seven, selecting whether the internal shift register architecture in the CPLDs, shown in Figure 2, loads or shifts data at the next edge of the 53 MHz system clock.

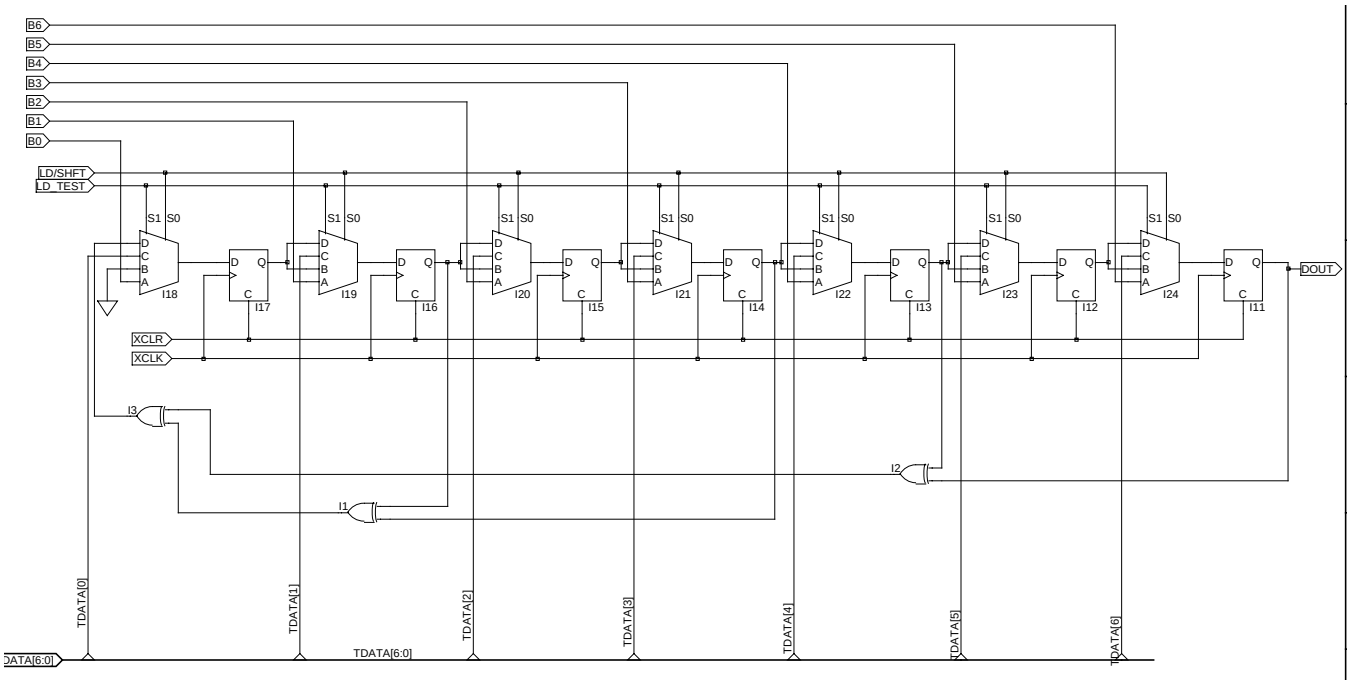


**Figure 2**

When the shift register is in the 'load' mode the data which is loaded may be either the data from the analog conversion circuitry (LD_TEST low) or a test pattern value from an internal register in the CPLD (LD_TEST high). The internal register may be loaded at any time from the on-board microprocessor, and a unique register exists per CPLD. The MSB of data from either source is immediately made available when the data is loaded, and the six following bits appear on the six following clocks. Since the load is synchronous all bits are valid for the same amount of time.

## *CPLD map to LVDS links and Test Patterns*

Each one of the 21-bit LVDS links on the 75 8-MCM AFE boards is driven by exactly two CPLDs. Bit 20 (MSB) is connected to a copy of the crossing clock (high for one tick of the 53MHz, then low for six ticks), used for link synchronization at the receiving end. The remaining 20 bits are broken into two 10-bit sections, each driven by one CPLD. Exactly which bit of the analog data goes to which bit of the LVDS links at what time slice has not yet been determined, as this affects the design of the Mixer box. However, the general rule followed is that all 64 bits from a given MCM all are sent to a given 10-bit segment (or half) of an LVDS link.

Some sharing of bits between CPLDs is allowed. Figure 3 shows the organization of data flow in the AFE. Each of the eight PLDs which feed the LVDS links have access to a total of 80 bits of information every beam crossing – 72 from the MCM and eight from the local microprocessor. In normal usage only 64 of the 72 MCM bits are used, and only six of the eight bits from the microprocessor are used, such that each PLD provides 70 bits of data to each link every 7 ticks of the 53 MHz clock – exactly using up the 10 bit bus provided.

It is envisioned that all of the data will simplistically flow left-to-right. However, a secondary bus structure also exists between the PLDs which allows 10 bits of communication between adjacent PLDs in either direction. These buses exist to allow for the possibility that part of the information intended for one LVDS cable may, for purposes of trigger formation, be better sent over a different cable; for instance, to minimize backplane transfers in the Mixer. It's useful to remember that this bus exists, but the presumption for the remainder of this document is that it is not used.

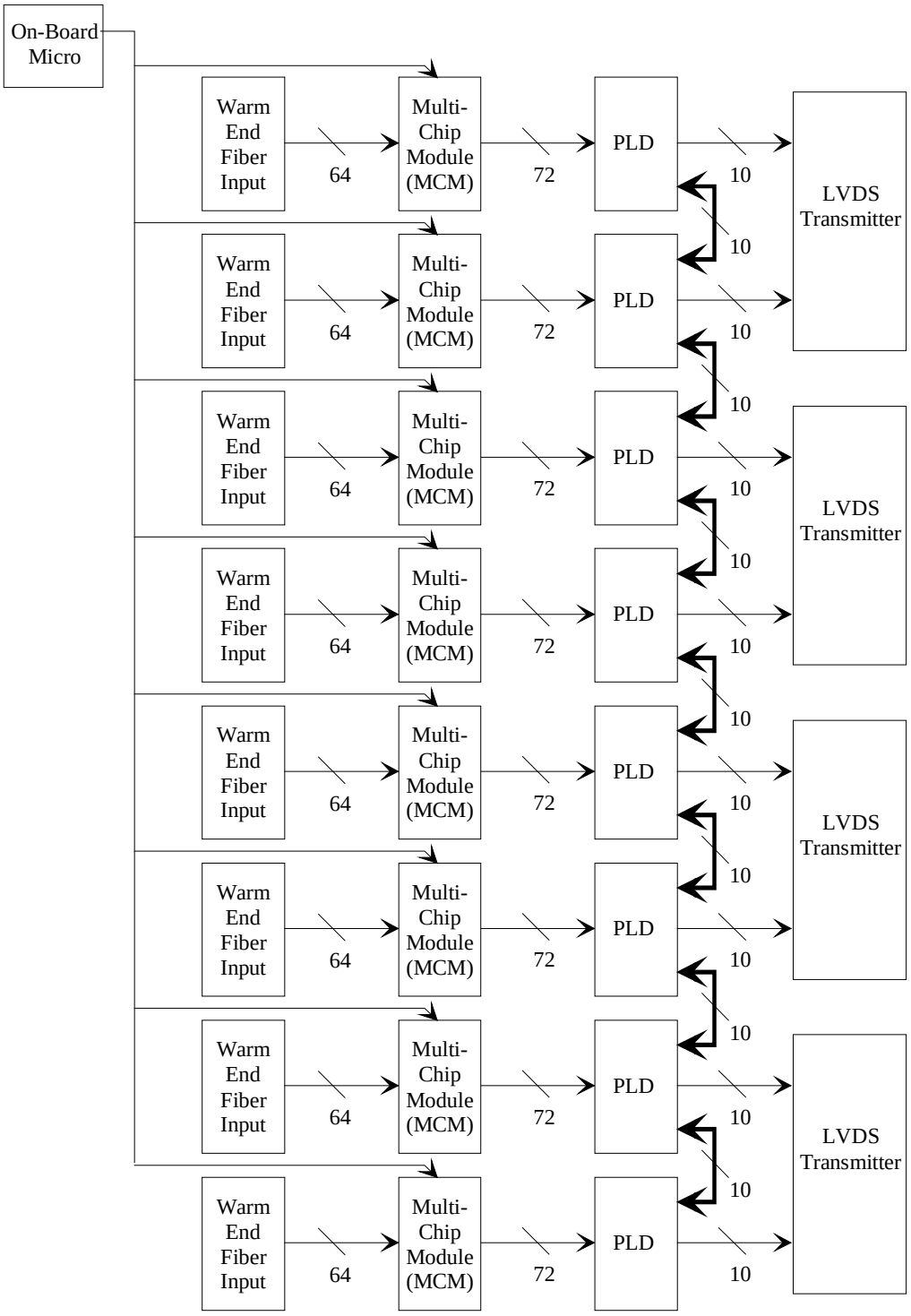


**Figure 3**

The effect of this bus structure when combined with the test pattern generation logic in the PLDs is that a psuedo-random number generator is formed. Based upon a 'seed' value loaded by the microprocessor at initialization time the PLDs create a sequence which drives all bits to 1 or 0 over a length of a few words. A copy of the pattern can be calculated in software; thus, if the data as received is correlated to the data as calculated, individual bit errors may be identified. Analysis of the pattern of mismatches over a reasonably short collection of words can be used to determine if the error is in the cable or in either the transmitter or receiver side. By then swapping cables between a functioning and non-functioning link at the receiver, the determination of which end of the cable is at fault may be made. By letting the pattern run for longer periods analysis of shorts between bits can also be provided.

An example of the patterns which may be generated is shown in Table 1. Three different seed values and the pattern generated by each seed are given as both binary and hex values.

| Seed value = 0x40 | | Seed value = 0x55 | | Seed value = 0x0C | |
|---|---|---|---|---|---|
| LVDS WORD OUT | Hexequiv | LVDS WORD OUT | Hexequiv | LVDS WORD OUT | Hexequiv |
| M L | | M L | | M L | |
| 0 0 1 0 0 0 0 0 0 1 | 81 | 1 1 1 1 0 1 0 1 0 1 | 3D5 | 1 0 0 0 0 1 1 0 0 0 | 218 |
| 0 0 0 1 0 0 0 0 0 0 | 40 | 0 0 0 1 1 0 1 0 1 0 | 6A | 0 1 0 0 0 0 1 1 0 0 | 10C |
| 0 0 0 0 1 0 0 0 0 0 | 20 | 1 1 1 0 1 1 0 1 0 1 | 3B5 | 0 1 0 0 0 0 0 1 1 0 | 106 |
| 1 0 0 0 0 1 0 0 0 0 | 210 | 1 0 0 1 0 1 1 0 1 0 | 25A | 0 0 1 0 0 0 0 0 1 1 | 83 |
| 0 0 0 0 0 0 1 0 0 0 | 8 | 0 1 1 0 1 0 1 1 0 1 | 1AD | 0 0 1 1 0 0 0 0 0 1 | C1 |
| 0 1 0 0 0 0 0 1 0 0 | 104 | 1 1 0 1 0 1 0 1 1 0 | 356 | 0 0 0 1 1 0 0 0 0 0 | 60 |
| 0 0 0 0 0 0 0 0 1 0 | 2 | 0 0 1 0 1 0 1 0 1 1 | AB | 1 0 0 0 1 1 0 0 0 0 | 230 |
| 1 0 0 0 1 1 0 1 0 1 | 235 | 1 1 0 1 1 0 1 1 0 0 | 36C | 1 1 0 1 1 1 1 1 0 1 | 37D |
| 1 1 0 1 0 1 1 0 1 0 | 35A | 1 0 0 0 1 1 0 1 1 0 | 236 | 1 0 1 1 1 1 1 1 1 0 | 2FE |
| 0 0 0 0 0 1 1 0 0 1 | 19 | 1 1 1 0 1 0 0 0 1 0 | 3A2 | 0 0 0 1 0 1 1 0 1 0 | 5A |
| 1 0 0 1 0 0 1 1 0 0 | 24C | 0 1 1 0 0 1 0 0 0 1 | 191 | 0 1 1 0 1 0 1 1 0 1 | 1AD |
| 1 0 0 0 1 0 0 1 1 0 | 226 | 1 1 1 1 0 0 1 0 0 0 | 3C8 | 0 0 0 1 0 1 0 1 1 0 | 56 |
| 0 1 0 0 1 0 0 1 1 1 | 127 | 1 0 1 0 0 1 1 1 0 1 | 29D | 0 0 1 1 0 0 1 1 1 0 | CE |
| 0 1 0 1 1 0 0 1 1 1 | 167 | 1 1 1 1 1 1 0 1 1 1 | 3F7 | 1 0 0 1 0 0 0 0 1 0 | 242 |
| 1 0 0 1 0 0 0 1 1 1 | 247 | 1 1 1 1 0 0 0 0 1 0 | 3C2 | 0 1 0 1 0 0 0 1 0 0 | 144 |
| 0 1 0 1 0 1 0 1 1 1 | 157 | 0 1 0 0 0 1 1 0 0 0 | 118 | 0 1 1 1 0 0 0 1 1 1 | 1C7 |
| 1 1 0 1 0 1 1 1 1 1 | 35F | 0 1 1 0 1 1 0 1 0 1 | 1B5 | 1 0 1 0 0 0 0 1 1 0 | 286 |
| 1 0 0 1 0 1 1 0 1 1 | 25B | 1 0 0 1 1 0 0 0 1 1 | 263 | 0 1 1 1 1 0 0 1 1 0 | 1E6 |
| 0 0 0 1 0 1 1 0 0 1 | 59 | 1 0 0 1 0 0 1 0 0 0 | 248 | 0 1 1 1 0 1 0 1 1 0 | 1D6 |
| 1 1 0 1 1 0 1 1 0 0 | 36C | 0 1 0 0 1 0 0 1 0 0 | 124 | 0 0 1 0 1 0 1 0 1 1 | AB |
| 0 0 0 0 1 1 0 1 1 0 | 36 | 0 1 0 0 0 1 0 0 1 0 | 112 | 1 0 1 1 0 1 0 1 0 1 | 2D5 |
| 1 1 0 0 1 0 1 1 1 1 | 32F | 0 0 0 0 1 1 0 0 0 0 | 30 | 1 1 0 0 0 0 1 1 1 1 | 30F |

A special case of the particular circuit shown is that for a seed value of 0, the data is always 0. In most cases every bit has been toggled once within seven words. Since the Mixer design is assumed to have a pipeline no more than seven words deep, this suggests that the Mixer system can be nicely tested by simply having the Mixer save the first seven words seen by a diagnostic dump and read them out. Another feature of the algorithm is that the seed itself is contained within the first seven words not once, but twice. The pattern'across' the first word always contains the seed, as does the value of the LSB over the first seven words. Thus, software can be 'smart' and perform a double-check of the seed value from the data itself. This feature allows the Digital Front End board to detect the seed 'on-the-fly' and perform in-system checks whenever desired.